

DRUKWERK

PORT BETAALD
LEIDEN

68KARANT

jaargang 1, nummer 7, mei 1985

afzender: 68000 gg, abrikozenstr. 86, 2564VT Den Haag

PORT BETALD
LEIDEN

08-11-1900



BOOK ARANT

Library of the University of Leiden

Dankzij 3 stukken van Ronald van de Kamp en een artikel van Dhr. H. Goeman (docent informatica Leiden) toch deze keer 6 blaadjes 68KArant. Bij deze daarvoor hartelijk dank. Ik had gehoopt een verhaal te hebben van Dhr. van 't Hul uit Utrecht, die mij vorige week belde dat hij een c't 68K-systeem uit Duitsland mee had genomen. Hij had ook nog informatie en een prijslijst over een implementatie van OS-9/68k meegekregen. Hij bleek echter voor een maand het ziekenhuis in gegaan te zijn. Van hieruit van harte beterschap toegewenst. Door de telefoon had ik een bedrag van 1145 mark vernomen, wat relatief goedkoop is, en ook dat het nog wel wat storende fouten bevatte. Op de Hannover Messe zou het gedemonstreerd worden.

bijeenkomst 4 mei.

Vanaf 11 uur in de technische school de Bron te Utrecht in een lokaal op de eerste etage. De school is gelegen aan de vaderrijndreef. Zie voor een route beschrijving een oude hccnieuwsbrief.

van 13-14 uur is er een plenair gedeelte.

Dhr. W. van Spronsen hoopt een draaiend c't systeem te kunnen tonen. In eprom is daarbij de taal PEARL met O.S. geleverd ipv. de taal FORTH.

Ronald van de Kamp komt met z'n MMU-kaart en zelf neem ik een in aanbouw zijnd 68k systeem uit het blad mc mee.

Verder hoop ik dat er een aantal mensen naar mij toe zullen komen, die iets voor de KArant hebben geschreven of willen gaan schrijven.

Komen er nog mensen die op de Hannover Messe zijn geweest?

Daar moet een nieuwe machine van Atari 'a la Macintosh maar dan veel goedkoper en een nieuw apparaat van textronics met de "SMALLTALK" programmeer omgeving te zien zijn geweest.

De volgende bijeenkomst is geprikt op zaterdag 31 augustus.

Ik hoop dan een 3D grafisch programma in pascal geschreven op de Macintosh te kunnen demonstreren.

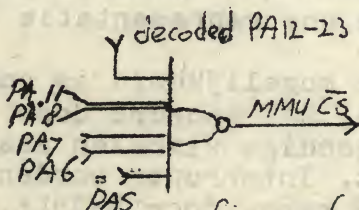
Tom den Duijf

KOPIJ

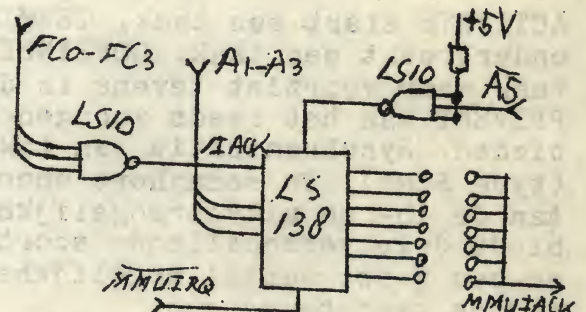
De 68KArant wordt gemaakt voor maar ook door 68000 gebruikers en geïnteresseerden. Klim ook eens in de "pen". Tekeningen bij voorkeur in zwart of rood maar niet in blauw evt. wel op licht blauw ruitjes papier, wat bij het kopiëren wegvalt. Redactie 68KArant: p/a Tom den Duijf, Morsstraat 50, 2312BN Leiden. tel. 071-134371.

Inhoud

PEARL.....	4
QL.....	5
MMU-hard.....	7
MMU-soft.....	9



figuur 6.



figuur 4

(bij MMU-hard)

PEARL

De programmeertaal PEARL (Process and Experiment Automation Realtime Language) is een universele modulaire programmeertaal die de belangrijkste elementen van hogere programmeertalen zoals pascal combineert met mogelijkheden voor realtime toepassingen en voor (quasi) parallelle programmering door meerdere taken gelijktijdig in uitvoering te nemen.

De ontwikkeling van PEARL begon in de Bondsrepubliek Duitsland in 1969.

Er zijn twee versies internationaal genormeerd: een kleinste versie als basis-PEARL en in volledige omvang als full-PEARL. Een officiële Pearl-vereniging ziet toe op de naleving en toekomstige aanpassing van deze standaards. PEARL biedt vele mogelijkheden om programma's gestructureerd vanuit deelconcepten samen te stellen. Programma's kunnen in modules opgedeeld worden. Modules kunnen taken, procedures en data bevatten. Taken en procedures kunnen op hun beurt data, procedures, blokken en elementaire opdrachten omvatten. Blokken kunnen wederom bestaan uit data, procedures, blokken en elementaire opdrachten. Modules kunnen naast een implementatiedeel ook een systeemdeel hebben. Zij kunnen afzonderlijk ontwikkeld en gecompileerd worden. De basistypen in de taal zijn:

FIXED, FLOAT, CHARACTER, BIT, CLOCK, DURATION en SEMA.

FIXED is vooral bedoeld voor gehele getallen, CHARACTER voor strings en textverwerking, BIT voor 0-1 rijen, CLOCK en DURATION voor tijdsaanduidingen, SEMA is bestemd voor synchronisatie van taken. Deze standaardtypen zijn nog in verschillende varianten aanwezig waardoor een grotere of kleinere nauwkeurigheid, lengte of bereik kan worden gespecificeerd. Uit deze basistypen zijn gestructureerde typen te maken, nl. velden (te vergelijken met (statische) array's) en structuren (te vergelijken met records). Zulke structuren kunnen ook via pointers (REF) benaderd worden. De procedures hebben de gebruikelijke parametermechanismen te vergelijken met by value en by reference. In verband met de multitasking mogelijkheden zijn de procedures alle herbetreedbaar (reentrant).

Een taak heeft een prioriteit. Die uitvoerbare taak die de hoogste prioriteit bezit heeft steeds voorrang boven de uitvoerbare taken met lagere prioriteit. Operaties op taken zijn:

ACTIVATE, TERMINATE, SUSPEND, CONTINUE, RESUME en PREVENT.

ACTIVATE start een taak, TERMINATE beindigt een taak, SUSPEND onderbreekt een taak, CONTINUE hervat een taak, RESUME onderbreekt een taak maar voorziet tevens in de hervatting ervan (op een later tijdstip) PREVENT kan het reeds voorgenomen starten of hervatten van een taak verbieden. Synchronisatie van taken geschiedt met behulp van semaphoren (type SEMA), de semaphore operaties zijn REQUEST en RELEASE. Aan de in- en uitvoermogelijkheden is bijzondere zorg besteed. PEARL biedt drie verschillende soorten van dataoverdracht met randapparatuur en een groot aantal mogelijkheden om data van de ene representatie in de andere over te voeren.

Ten slotte dient nog vermeld te worden dat er de mogelijkheid is om op onvoorziene gebeurtenissen (event's) te reageren. Dit gebeurt door middel van interrupts welke niet alleen via uitwendige signalen maar ook in het programma zelf kunnen worden opgewekt. Interrupts worden uitgeschakeld of weer mogelijk gemaakt door DISABLE resp. door ENABLE. Uit dit korte overzicht moge duidelijk zijn dat PEARL een bijzonder fraaie taal is die door zijn extra mogelijkheden op het gebied van realtime processing en multitasking zeer zeker de moeite waard is.

QL.

Er is inmiddels weer enige tijd verstreken sinds ik voor het laatst rapporteerde over de QL. Zo is er heel wat nieuws te berichten, en wel:

A. Software.

Voor schaakliefhebbers is er een 3-D chess programma (QL-chess) dat, naar men zegt, best redelijk schaakt. Dit programma heeft ook de mogelijkheid om het totale verloop van het spel op te slaan. Je kan dus de machine afzetten en later verder gaan.

Op het gebied van Assemblers het volgende:

De nieuwste update van de Metacomco assembler is ontvangen. Bijna alle bezwaren die ik tegen dit pakket had zijn hierbij opgeheven. B.v. de default drive hoeft niet meer 'MDV1' te zijn; de editor kan als task in het geheugen blijven; de assembler kan ook als task blijven draaien, maar helaas is er in de standaard QL te weinig geheugen om beide op te bergen. De assembler blijft ook in de nieuwe versie toch nog overlay's laden vanaf de (gekozen) default drive. Dit is jammer. Wat wel handig is, is dat de windows van editor, assembler en linker bij de initialisatie kunnen worden ingesteld. Het allerbeste van deze nieuwe update is de aanwezigheid van een Linker. Nu kan tenminste fatsoenlijk in assembler gewerkt worden met externe referenties en definities. De prijs is inmiddels gezakt tot # 40,-.

De firma TDI in Bristol levert het bekende UCSD-p systeem met Pascal en Fortran 77 tesamen met een Development Toolkit. Bevat prachtige zaken zoals daar zijn: Symbolic Debugger en 68000 Native Code Generator. De kosten voor het totaalpakket zijn # 150,-. Naar het schijnt komen de resultaten van de benchmarks overeen met die van de IBM-AT qua snelheid!

De tweede versie van de gebundelde Psion software is gesignaleerd bij nieuw gekochte QL's. De Rom's zijn ook veranderd.

De nieuwe versie van het QDOS-CS (in ROM) heeft er een aantal Basic commando's bij gekregen (b.v. ON ERROR GOTO) en een aantal lastige fouten in de conversie utilities zijn verdwenen.

Voor de prijs van slechts #100,- is sinds april ook een APL te krijgen (met 32K plug-in ROM) dat IBM mainframe APL compatible is. Dit QL-APL kan in mnemonics werken, maar ook de echte APL symbolen gebruiken, verwerken en weergeven. Een masker voor het toetsenbord wordt bijgeleverd.

Alhoewel je prima assembler programma's kunt ontwikkelen op de QL, is het debuggen soms een probleem. De firma Digital Precision in Londen brengt een Monitor op de markt voor # 20,-. Debuggen kan gedaan worden op code niveau met vele commando's waaronder jump / trace / set / show / dump / set regs / hex / en dergelijke. Een disassembler wordt bijgeleverd.

Voor spelletjes e.d. zijn er de volgende zaken:

- een SPRITE generator die machinecode levert,
- een Backgammon spel,
- een adventure game (The lost kingdom of Skull),
- nog zo iets (The Wild-West),
- een QL-art drawing program met zoom, windows, paintbox, brush, turtle-graphics,
- een war game (D-day),

nog een grafisch programma (GRAPHIQL) dat in low-resolution werkt met 8-kleuren per pixel, een airbrush en veel meer (# 35,-),

een vliegveld simulator (Radar traffic controller) (# 11, -.)

B. Hardware.

De productie van peripherals begint nu goed op gang te komen, hetvolgende is nu verkrijgbaar:

1. modem QCOM dat V21/V23 met 75/1200, 1200/1200, 300/300 instelbaar is en bestaat uit 3 delen
 - a. interface QL seriele port naar modem,
 - b. auto-dial / answer unit,
 - c. modem zelf met directe connectie naar telefoon lijn.

Bijgeleverde software voor VT52 en VT100 emulatie en file-transfer.

De prijs voor de complete set is mij niet bekend, maar ligt wel hoog.

2. een QL Compak modem voor # 160,-. met ingebouwde Centronics uitgang. Snelheden 300/300, 1200/75, 1200/1200. Enige software gratis. Wat dit bevat is onbekend, er schijnt een terminal emulator bij te zitten.

3. Memory expansion.

- a. een serie 64, 1228, 256K voor resp. #75, 115, 150 (ex BTW) van de firma PCML.
- b. een serie 64, 128, 256, 512K voor resp. #100, 130, 170, 260 van de firma Medic Data Systems.
(de 512K heeft een eigen voeding nodig.)
- c. Een serie 64, 128, 256, 512K voor #115, 185, 349, 579 van de firma Quest
- d. een kaartje 256K voor, #200,-. uitbreidbaar tot 512K
(met 8*256Kb chips) van de firma Simplex.

4. Systeem expansie.

De firma Miracle Systems verkoopt een expander board, zodat 2 kaarten tegelijk in de QL geplukt kunnen worden.

Voor zover mij bekend zijn alle extra boards van de diverse fabrikanten zo ingericht dat de expansie connector van de QL niet wordt doorgevoerd. Dat hindert de uitbreidbaarheid wel! Dit simpele kaartje moet maar liefst # 40,-. kosten.

5. Disk drives.

- a. Q Disk Interface voor #150,-. Kan twee floppies aan (5 en/of 3 inch) in single of dubbele density met 40 of 80 tracks.
- b. CST disk interface. #115,-. Kan hetzelfde en wordt geleverd met software voor microdrive emulatie.

De QL standaard voor disks is 9 sectors/track en 512 bytes/sector. Dus voor $2 \times 680 + 600 = 1960$ gulden heb je 1,5 Mbyte aan opslag. (dual 3 1/2 " 80 track DD + interface)

- c. Interface van Medics Data Systems is gecombineerd met hun geheugen uitbreiding.

6. Kaarten met andere processoren.

De firma PCMC heeft een Z-80 CPM 3.0 kaart aangekondigd: geplande levering vanaf juli 1985. Prijs onbekend.

7. De firma Care Electronics levert EPROM cartridges, die ingeplukt kunnen worden in de ROM expansie port.

8. Boeken

Er zijn diverse QL boeken verschenen. Het enige boek dat de moeite waard is om gekocht te worden blijft 'QL Advanced User Guide' van A. Dickens. De rest geeft enkel ernstige boekenkast vervuiling..

Ronald van der Kamp

Zo ongeveer eind mei hoop ik het c't 68 bouwpakket uit Duitsland te ontvangen. Ik heb besloten om in de tussentijd het reeds lang geleden geplande memory management board te gaan wire-wrappen. De bedoeling van dit boardje is dat het inpluigt (mbv. 64-pin DIP flatcable connector) in de 68000 voet. Alle benodigde onderdelen heb ik kunnen vinden, behalve de delay chip DDU-4-5250, die vertragingen van 20/40/60/80 nanosec maakt en de 64-pin DIP connector die in de 68000 voet past. Tips hieromtrent worden gaarne in ontvangst genomen!

Aangezien de MMU 68451 slechts zo'n 32 paging segmenten (descriptors) bevat is mijn kaart zo gekonstrueerd, dat via een DIN-AC-64 pin en een 16 pin DIP connector, een tweede kaart met maximaal 5 MMU's kan worden bijgeschakeld.

Het schema dat ik gebruik is terug te vinden in de Motorola Application Note AN-880. Het globale schema staat getekend in figuur 1. De datalijnen D0-D15 van de processor worden naar het systeem doorgegeven. Ook de address lijnen A1-A7 gaan gewoon door. De lijnen A8-A23 (16 stuks) gaan naar de address input lijnen van de MMU.

De MMU heeft twee aspecten: ten eerste is hij een gewone (echte) 68000 peripheral, ten tweede is hij een vertaler van de logische (software) adressen naar de reële fysieke adressen.

1. peripheral chip.

De MMU heeft een chip select pin. Op mijn kaartje wordt het signaal als volgt uit de addresslijnen gemaakt:

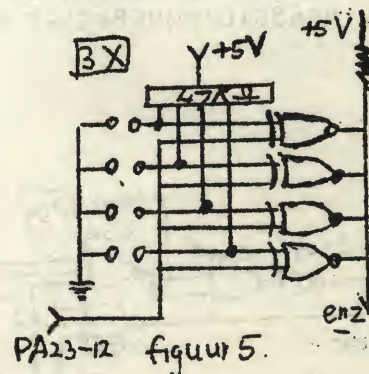
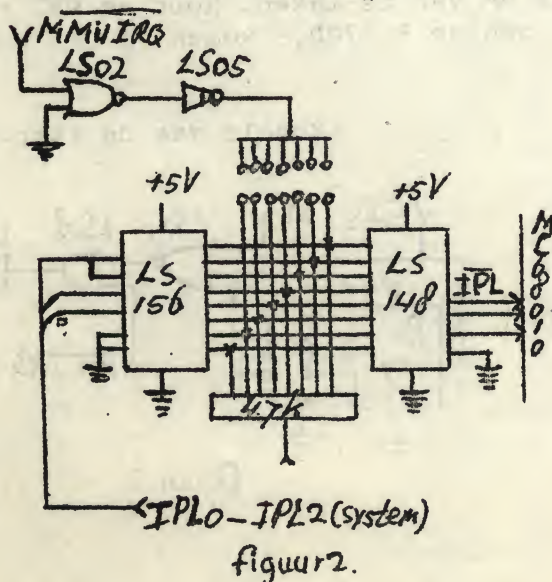
De twaalf hoogste vertaalde adreslijnen PA23-PA12 worden vergeleken met een twaalf-tal ingestelde toestanden (zie figuur 5). Het resulterende signaal wordt samengeknoopt met PA11-PA8, de onvertaalde A7 en A6 en de gemaakte fysieke address strobe /PAS. De addresslijnen A1-A5 fungeren als register selects voor de MMU (inputs RS1-RS5). Zodoende neemt de MMU een 4K plaats in in het virtuele addressbereik. De MMU registers zitten in de bovenste 32 bytes van het 4K address blok.

Om pootjes aan de chip te sparen zijn de 16 uitgaande addresslijnen en de 16 bidirectionele datalijnen gemultiplext. Dit vergt 2 stuks 74LS245 voor de data lijnen en twee 74LS373 voor de uitgaande adressen. De MMU geeft de juiste signalen om de toestand van deze 4 chips te zetten. (signalen /ED en /HAD)

De /DTACK kan gewoon met de systeem lijn verbonden worden.

De MMU interrupt request /IRQ wordt als volgt behandeld: (figuur 2)

de drie van het systeem binnenkomende interrupt priorities /IPL0-/IPL2 worden gedecodeerd tot de 7 interrupt levels mbv. een 74LS156 dual 2 to 4 decoder, die geschakelt wordt als een 3 naar 8 decoder. Met behulp van een simpel jumpertje kan de MMU /IRQ met de gewenste interrupt lijn verbonden worden. Nu vindt weer een decode plaats met een 74LS148 en zijn de /IPL0-/IPL2 weer terug. Zo is de MMU /IRQ 'onzichtbaar' voor het systeem. Het is de bedoeling dat alle acties van de MMU zo veel mogelijk onzichtbaar blijven.



Met een interrupt acknowledge cyclus wordt door de processor het niveau van de beantwoorde interrupt op de lijnen A1-A3 gezet (en de rest van de adressen is hoog). Wanneer een Iack cyclus voor de MMU plaats vindt dan moet dit onzichtbaar blijven voor het systeem. Dit gaat als volgt: (figuur 4) Het /MMUIACK signaal wordt mbv. dipswitches uit hetzelfde niveau als de MMUIRQ gefiltert. Uit de A1-A3 wordt het level gedecodeerd met een 74LS138 (3 to 8 decoder) met behulp van de signalen /AS, /MMUIRQ en /IACK. De /IACK wordt gemaakt uit de function codes FC0-FC2. Een gebufferde /IACK wordt ook gebruikt om de /MAS en /ALL lijnen van de MMU omhoog te brengen. Deze constructie heeft de volgende reden: Gedurende de Iack cyclus is de /AS valid, zodat de MMU denkt het address te moeten vertalen. Dit kost je dus 1 segment voor een address dat toch nergens voor gebruikt wordt in het systeem. Door de /MAS en de /ALL in dit geval laag te leggen (mbv. open collectors) zal de MMU denken, dat een andere (parallel geschakelde) MMU de vertaling op zich neemt. Voor vele systemen is het ongezonder om de /UDS en de /LDS te krijgen, wanneer geen activiteit van het systeem gevraagd wordt op de data bus. In figuur 7 is het schema voor de generatie van de generatie van de /PUDS en de /PLDS gegeven. De /PLDS is onmogelijk gedurende een interrupt acknowledge van de MMU; immers de MMU plaatst de interrupt vector op de data lijnen D0..D7.

De uiteindelijke address strobe /PAS moet ook niet het systeem in gaan gedurende een MMM Iack. (zie figuur 3).

De generatie van de nieuwe /AS wordt gedaan met /MAS, /AS en de vertraagde /MAS. Deze vertraging kan noodzakelijk zijn vanwege de setup tijd van de adressen in het systeem. (wie weet de address set-up time voor de c't 68 ??)

De /PAS wordt niet doorgegeven naar het systeem gedurende de MMU Iack cyclus.

2. memory management.

Bij normale werking (address vertaling) kunnen devolgende zaken gebeuren:

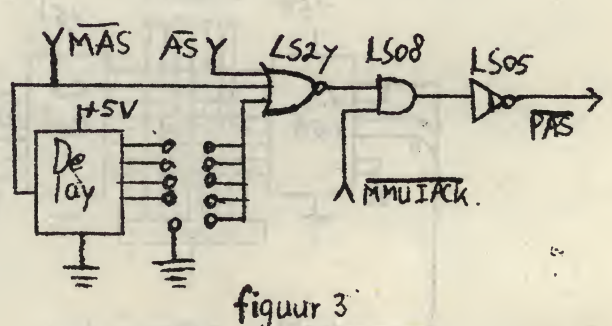
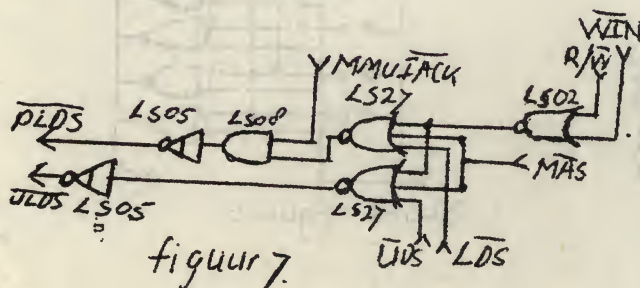
1. de MMU signaleert een interrupt. De hardwarematige afhandeling is hierboven besproken.
2. alles gaat goed. /MAS wordt gegeven.
3. als het onmogelijk is het binnenkomende address te vertalen treedt een PAGE FAULT op. Dit is een Bus Error voor de MC.
4. address is vertaalbaar, maar de actie is een write op een write-protected address.

De /WIN (write inhibit) zorgt ervoor dat (b.v. tijdens een TAS-instructie) er ook hardwarematig geen write kan plaats vinden door de /PLDS en de /PUDS te blokkeren. (zie figuur 7).

Tot zover dit nogal technische verhaal. Als ik dit kaartje met al zijn draadjes en chipjes werkt, en er is voldoende belangstelling, dan hoop ik ooit een net tweezijdig eurokaartje er van te maken. Houd er wel rekening mee, dat de kosten van zo'n kaart toch op F. 700,- komen.

(n.l. MC68010+MMU68451=F 600.-)

Ronald van de Kamp.



De MMU vanuit software oogpunt.

De MMU registers vallen in 2 groepen uiteen:

1. de 32 descriptors, die elk een zgn. memory segment beschrijven,
2. de systeem registers.

Een descriptor ziet er als volgt uit:

- a. Logical Base Address. (LBA). Een 16 bits register dat tesamen met LAM de logische address range van dit segment definieert.
- b. Logical Address Mask. (LAM). Dit is een 16 bits mask. Nullen geven bitposities aan in de LBA die moeten worden 'meegenomen' in de address matching.
- c. Physical Base Address. (PBA). Bevat een 16 bits address, dat samen met LAM en het binnenkomende address, het fysiche address vormt.
- d. Address Space Number. ASN is een 8 bit nummer dat, samen met de ASM een matching verzorgt met het address space nummer dat (geindexed door FC0..FC2) in de Address Space Table staat.
- e. Address Space Mask. ASM is een 8 bit masker. Nullen geven een 'doet er niet toe' in de corresponderende bitposities van ASN bij de address space match.
- f. Segment Status Register (SSR) bevat de volgende status bits:
 - U. Used flag. Wordt gezet als er toegang tot het segment is geweest. (translate)
 - I. Interrupt control bit. Als je deze zet dan heeft het gebruik van deze descriptor een interrupt tot gevolg (indien interrupt enabled natuurlijk).
 - IP. Interrupt Pending. Wordt gezet als er al een interrupt is (I gezet).
 - M. Modified. Bij gebruik van de descriptor wordt deze gezet.
 - WP. Write Protected. Kan je zetten. Wordt er toch een write gedaan, dan is er een write violation.
 - E. Enable bit geeft aan of de descriptor mee doet in het matching process of niet.

De systeem registers.

- A. De Address Space Table (AST) is georganiseerd als 16 8-bits registers. De function code, die de MMU binnen krijgt wordt gebruikt als index voor deze tabel om zo een Address Space Number te selecteren. Er is zodoende een duidelijk onderscheid in User/Supervisor Data/Program en Iack cycli.
- B. De accumulators ACO..AC8.
 - In geval van read/load van descriptors bevatten deze : logical/physical Base Address, Logical Address Mask, Address Space Nummer, Segment Status Register en Address Space Mask.
 - Bij directe vertaling bevatten ze: Logical translation register, physical translation register en Address Space Number.
 - Bij een fault gedurende vertaling: logical address, cycle Address Space Number.
- C. Globale Status Register. (GSR). met de bits
 - F. Fault : de MMU kan niet vertalen.
 - DF. Double Fault .
 - IE. interrupt enable. De IRQ kan functioneren.

- D. Locale Status Register.(LSR).met de bits:
 RW. De status van de Read/Write pin bij het optreden van de fault,
 GAT. Global Accumulator for Translate. Wordt gezet als je een logisch inconsistente descriptor hebt gekregen.
 GAL. Global Accumulator for Load.
 LIP. Local Interrupt Pending voor als er een IP bit is gezet in een descriptor.
 L4..L7. fout code.
- E. Descriptor Pointer DP identificeert de descriptor die betrokken is bij lezen,schrijven of vertalen.
- F. Resultant Descriptor Pointer (RDP) is read only. Geeft aan welke descriptor betrokken was bij een write-violation of een fout bij het laden van een descriptor.
 Er bestaat een prioriteit tussen de descriptors, nummer nul heeft de hoogste prioriteit. Door herhaaldelijk RDP te lezen krijg je opeenvolgend de nummers binnen van de betreffende descriptors.
 De volgende bits zitten er in:
 NVR. no valid result,
 R0..R4 het descriptor nummer in de priority encoder.
- G. Interrupt Descriptor Pointer IDP is read only en dient om te bepalen welke descriptor(s) een interrupt hebben gegenereerd. Er is eenzelfde prioriteiten schema als voor de RDP. De bits zijn:
 NVI : er is wel/geen interrupt meer over in de priority decoder,
 I0..I4 : het descriptor nummer in de priority encoder.

Uit het bovenstaande overzicht blijkt duidelijk dat er voldoende hardware is om een serieus virtueel systeem op te bouwen. Het is wel noodzakelijk om een MC68010 te gebruiken, omdat deze processor voldoende informatie over zijn interne toestand op de systeem stack gooit wanneer er een bus-error optreedt. De instructie kan zodoende later zonder moeilijkheden vervolgt worden. Bij een page-fault heeft de MMU het betreffende logische address, het Address cycle Space Nummer en de status van de read/write pin in de accumulators gelatched. De software van de page-fault handler kan de fault repareren (bv. door de benodigde pages vanaf disk naar het RAM geheugen te transporteren en een descriptor in de MMU te zetten) en de vastgelopen instructie doen vervolgen.

Vaak heb je geen ruimte meer in RAM om een page vanaf de disk te plaatsen. Dan moet er bekeken worden welke de beste page is om weg te doen. Elke descriptor heeft een Used en een Modified flag, die je ook zelf kunt zetten. Met wat administratieve overhead en een timed interrupt is het mogelijk om inzicht te hebben in welke descriptors laatstelijk veel in gebruik zijn geweest, en welke reeds lange tijd in rust zijn. Als je er van uit gaat dat een descriptor die lange tijd ongebruikt is, ook in de toekomst voorlopig niet gebruikt zal worden, dan zijn er enkele efficiënte en simpele page-replacement algoritmen mogelijk.(b.v. Working-Set Clock) Aangezien elke descriptor in de MMU een addressspace nummer en mask heeft, is het mogelijk om elke task zijn eigen (unieke) address space te geven; elke job kan zijn eigen mapping-table hebben. Task switching wordt ook simpel. Bv. het starten van een user job:

schrijf AST1 en AST2 (user data/program address space) met het 8-bit address space nummer van deze user task. Zet het status register (met de supervisor bit cleared) en de program counter voor de user-task op de systeem stack en doe een RTE instructie. Gevolg: processor in user mode en de logische adressen worden gemapt mbv. AST1&2.

doe een MOVEM predecrement gevolgt door een illegal instruction. De MC68010 heeft dan beide instructies in de pipe zitten, MOVEM verandert AST5 en AST6, dan komt een trap naar de illegal instruction handler, waarin bekeken wordt wat de illegal instruction was (je zet dus een job-switch code in de ill. instr.), waarna door getrapt wordt naar de nieuwe supervisor task.

De 68010 kent een zgn. loop-mode.: als een DBcc instructie met een displacement van -4 wordt gevonden , en de voorafgaande instructie is 'loopable', dan hoeven er, zolang als de loop loopt , geen instructies meer binnen gehaald te worden bv.

DBEQ DO, LOOP

```
ADD [BWL] (Ax),Dy
CMP [BWL] (Ay)+,(Ax)+
CMPA[BWL] -(Ay),Dx
ASL [BWL] #1,(Ay)
```

Voor vragen, opmerkingen, tips , wetenswaardigheden e.d. sta ik open.

The diagram illustrates the MMU 68451 interface between the MC 68010 and MC 68000. The MC 68010 (left) provides signals including \overline{DTACK} , \overline{AS} , \overline{BERR} , \overline{LDS} , \overline{LDS} , $\overline{IPL0-IPL2}$, and \overline{RESET} . The MC 68000 (right) provides $\overline{Do-15}$, \overline{PUDS} , \overline{PLDS} , \overline{PAS} , \overline{voet} , $\overline{A-23}$, \overline{M} , \overline{C} , $\overline{68000}$, and \overline{voet} . The MMU 68451 (center) has inputs for \overline{DTACK} , \overline{AS} , \overline{IRQ} , $\overline{RS1-RS5}$, \overline{FAULT} , \overline{CS} , and \overline{RESET} . It also has outputs for $\overline{R/W}$, \overline{ED} , $\overline{PADO-15}$, \overline{WIN} , \overline{LDS} , \overline{LDS} , \overline{HAD} , and $\overline{A1-A7}$. The MMU 68451 is connected to the LS245 2X buffer (top) and the LS373 2X buffer (bottom). The LS245 2X buffer is connected to the MC 68010 and the MC 68000. The LS373 2X buffer is connected to the MC 68000 and the MC 68010. The MMU 68451 also contains a PDS generator and a PAS generator. The PDS generator is connected to the MC 68000 and the MC 68010. The PAS generator is connected to the MC 68000 and the MC 68010. The MMU 68451 also contains a Priority inserter and an Address decode block. The Priority inserter is connected to the MC 68010 and the MC 68000. The Address decode block is connected to the MC 68000 and the MC 68010.

- 11 -

1. The first part of the report is a general description of the project and its objectives. It includes a brief history of the project and a statement of the problem to be solved.

2. The second part of the report is a detailed description of the methodology used in the study. It includes a description of the data collection methods, the statistical methods used for data analysis, and the experimental procedures used to test the hypotheses.

3. The third part of the report is a discussion of the results of the study. It includes a summary of the findings, a comparison of the results with previous research, and a discussion of the implications of the findings for future research.

4. The fourth part of the report is a conclusion and a list of references. The conclusion summarizes the main findings of the study and provides a final statement on the project. The references list the sources of information used in the study.

5. The fifth part of the report is a list of appendices. These include any additional information that is relevant to the study but is too large to include in the main body of the report.

